AMENDMENTS TO THE SPECIFICATION

Please replace the following paragraph(s):

Page 3, line 22 to Page 4, line 4:

Four related inventions are described herein. Embodiments of each invention improve aspects of communication among network equipment to improve, for instance, network configuration and management. All four inventions are described below in the context of a network of data switches. The inventions, however, are not limited to the illustrated embodiments, and are generally applicable to a wide variety of networks including, for instance, a local area network ~~local area network~~ (LAN).

Page 5, lines 12-18:

Various embodiments of the present inventions can be used to automatically ~~manage topology~~ manage the topology of the stack so that, for instance, configuring the stack topology to operate like a single large switch can be as simple as plugging in a few data cables, switches can be added or removed from the stack as needed with little interruption in service, etc. The advantages of a self-managing network of switches are numerous. For example, a user can simply couple the switches together in any random order, let the switches configure themselves, and begin using the stack.

Page 9, lines 1-8:

In block 310, the switch obtains a current value associated with a key. A key is used as an index in the distributed dictionary. In the context of topology management ~~manage~~, a key may be an identifier for a particular switch and an identifier for a particular intra-stack port on the switch. That is, a network topology can be defined in terms of switches and ports. The distributed dictionary may include an entry for every switch and port in the network that has a cable coupled to it. In which case, the value associated with each key may be an identifier for a neighbor switch and its port connected to the switch and port indexed by the key.

Page 9, lines 19-23:

Blocks 330 and 340 ~~comprises one~~ comprise one embodiment of registering an attribute. In block 330, the key/value pair is stored locally in the switch's distributed dictionary application, along with the incarnation identifier for the value. In block 340, the switch multicasts the attribute to the rest of the switches to be stored by the respective distributed dictionaries.

Page 12, lines 1-7:

In block 420, if the key matches a previously stored key, the switch checks in block 440 to see if the attribute has a different incarnation value. If it does not have a different incarnation value, then the attribute is likely just a refresh of an earlier received attribute. In the illustrated embodiment, the switch just returns to

block 410 to wait for the next attribute to arrive. In an alternate embodiment in which attributes have limited life spans, the ~~repeat attribute many~~ repeated attribute may replace the currently stored attribute or the switch may simply reset a time on ~~the attributes~~ the attribute's life span.

a6

Page 14, lines 15-18:

In block 605, a switch starts the process ~~by uses its~~ by using its media access control (MAC) address for its stack identifier. All of the switches in the stack will eventually adopt the same stack identifier in order to identify the stack to which they belong. In which case, the switch is likely to change its stack identifier later on.

a7

Page 15, lines 7-10:

In block 615, the switch broadcasts each of the stored attributes on a corresponding port. The broadcast is point-to-point, as opposed to multicast, and is intended to go no further than an immediate neighbor. That is, if the attribute reaches a neighbor, the neighbor need not forward the attribute ~~the attribute on~~.

a8

Page 15, line 20 to Page 16, line 2:

Switches cannot be adjacent to one another for networking purposes if they are not part of the same stack. So, they agree on the stack identifier before determining adjacencies for a variety of reasons. As discussed below, in one

a9

embodiment, the stack identifier corresponds to a master switch. The adjacencies are used by the master switch to determine a topology for the stack. If the master switch is yet to be designated, then ~~their is no need~~ there is no need to record an adjacency.

Page 17, line 21 to Page 18, line 6:

In block 710, the master switch accesses the set of adjacencies in the distributed dictionary as reported by all of the switches. In block 720, the master switch provides the set of adjacencies to a graph-theory algorithm. In one embodiment, a known shortest path first (SPF) algorithm is used. The algorithm operates on a set of nodes and links between nodes to determine the shortest path between any two nodes. In one embodiment, SPF ~~operates on the bases~~ operates on the basis of propagation delay through the respective network paths. In addition to determining shortest paths, SPF also ensures that paths do not loop back on themselves, which could potentially cause lost data or other problems.

Page 19, lines 16-22:

Basically, the fourth invention insures that all old links are disabled before new links are enabled. For instance, in the example above for the loop in Figure 2 ~~in Figure 3~~, if the old link between switches 120 and 130 were disabled before the new link was formed between 110 and 130, no transient loop would occur. In one embodiment, a port is disabled only if it cannot send and cannot receive

data, and it has no packets buffered and waiting for transmission. By including sending and receiving as requirements for disablement, a link can be disabled by disabling a port on just one end of the link.

a11

Page 23, lines 3-7:

In all of the embodiments of all the inventions described herein, alternate embodiments may not require all of the elements shown, may include additional elements, ~~and perform~~ and may perform one or more elements in a different order. Furthermore, even though the embodiments were illustrated in the context of a switch stack, the inventions are applicable to a wide variety of alternate network environments as well.

a12

Page 23, lines 8-15:

Figure 11 illustrates one embodiment of a hardware system intended to represent a broad category of network devices such as personal computers, workstations, ~~switches routers,~~ switches, routers, and/or embedded systems. In the illustrated embodiment, the hardware system includes processor 1110 coupled to high speed bus 1105, which is coupled to input/output (I/O) bus 1115 through bus bridge 1130. Temporary memory 1120 is coupled to bus 1105. Permanent memory 1140 is coupled to bus 1115. I/O device(s) 1150 is also coupled to bus 1115. I/O device(s) 1150 may include a display device, a keyboard, one or more external network interfaces, etc.

a13